



Animações com Javascript

Estamos de volta com muito JavaScript e animações!

Já aprendemos sobre HTML, CSS e programamos em JavaScript, bem como vimos como aplicá-los em nossos jogos. Sabemos desenhar tanto em bitmap, quanto em representações vetoriais dentro de nossas aplicações web. Agora vamos aprender como criar animações interativas com técnicas de programação aplicadas em animação com a linguagem JavaScript.

Passos para animação básica

Para o desenvolvimento de nossos jogos web, vamos precisar animar objetos na tela, como obstáculos para o jogador, os personagens e o próprio personagem do jogador. Isso pode ir um pouco além do que já aprendemos até aqui, pois o que vamos criar a partir de agora são animações com base na linguagem de programação JavaScript.

Isso quer dizer que o que aprendemos sobre animações com CSS e SVG são muito úteis, mas precisamos ir mais além para criarmos jogos completos, com ajuda da programação.

Nesta unidade trabalharemos em animações aplicadas aos objetos e *sprites* dentro da área de desenho do Canvas, mas o que será visto aqui pode ser aplicado a qualquer aplicação web e a qualquer objeto no DOM.

O método requestAnimationFrame()



Para criarmos nossas animações em JavaScript, usaremos o método *requestAnimationFrame()* do objeto *window*. Este método informa ao navegador que iremos realizar uma animação, executando blocos de código específicos antes que a tela seja atualizada (*repaint* da tela). Isso irá gerar quadros de animação em sequência, dada uma taxa de quadros ou frames por segundo (FPS).

O método tem como argumento uma *callback* que deve ser invocado antes da repaint. Vejamos a sintaxe básica:

```
window.requestAnimationFrame(callback);
```

Esta *callback* nada mais é do que uma função (ou método) a ser chamada do próximo *repaint* da tela do navegador. A taxa de frames por segundo é de 60 por segundo normalmente, no entanto existe uma recomendação da W3C para que essa taxa acompanhe a taxa de atualização do *display* do usuário.

Isso quer dizer que este método de animação se valerá das condições de GPU e de otimização disponíveis no dispositivo do usuário. Quanto melhor a performance de GPU, melhor será o desempenho do jogo. Além disso, esse método tem mais algumas vantagens quando aplicado em animações:

- O navegador otimiza sua implementação, assim as animações serão suaves;
- Animações em guias inativas irão parar, permitindo que o processador esfrie;
- Ajuda a otimizar o consumo de bateria.

Para escrever nossos programas, podemos omitir o objeto *window*, escrevendo somente o método diretamente. Assim, uma utilização básica pode seguir a seguinte estrutura:

```
// Função de callback a ser invocada pelo método de animação.
function animar() {
    // Aqui virá o bloco de código para
    // reposicionar objetos na tela.

    // Realizar uma nova atualização.
    requestAnimationFrame(animar);
}
// Chamar o método de animação.
requestAnimationFrame(animar);
```

Perceba que o método *requestAnimationFrame()* deve sempre ser invocado quando precisarmos atualizar nossa animação, por isso, ele deve ser chamado novamente dentro da função *callback*. Vamos a um exemplo mais prático?

Animando Objetos do Canvas

Para exercitarmos uma animação nesse resumo, vamos considerar a animação de um retângulo desenhado através da Canvas API.

Vamos antes para a composição do nosso documento HTML. Ele será simples, contendo apenas um elemento <canvas> quadrado. Dessa forma:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Animação com JavaScript</title>
  </head>
  <body>
    <canvas id="ui-canvas" width="300" height="300"></canvas>
  </body>
</html>
```



Vamos desenhar um retângulo dentro desse canvas e fazer uma animação simples, movendo-o para a direita, ou seja, nosso programa em JavaScript fará a atualização da posição no eixo X do retângulo.

Teremos então uma variável para controlar essa posição, vamos inicializá-la com o valor zero e a cada atualização do frame de animação, o valor dela será atualizado em 2 unidades. Para dar a noção de movimento, limparemos a área do canvas antes de desenhar o retângulo.



```
// Obter o contexto para acessar a API de desenho 2d
let canvas = document.getElementById("ui-canvas");
let ctx = canvas.getContext("2d");
// Posição inicial no eixo x.
let posX = 0;
// Função de callback a ser invocada pelo método de animação.
function animar() {
    // Limpar a área de desenho do canvas.
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    // Desenhar um retângulo.
    ctx.fillRect(posX, 10, 20, 20);
    // Atualizar a posição em x do retângulo
    // para a próxima atualização.
    posX += 2;
    // Realizar uma nova atualização.
    requestAnimationFrame(animar);
}
// Chamar o método de animação.
requestAnimationFrame(animar);
```

Pronto! Agora já temos uma noção de como criar animações dentro de nossos jogos web. Vimos aqui um exemplo junto com a API de desenho do Canvas, no entanto, esse recurso pode ser aplicado para animar qualquer elemento do DOM dentro do documento HTML. Ou seja, com isso podemos criar experiências muito ricas de interface.

Nas próximas unidades, veremos algumas bibliotecas JavaScript para nos ajudar a criar essas animações e experiências.

Até a próxima!

Atividade Extra



Acesso o link abaixo da MDN Web Docs, leia e pratique o exemplo contido no artigo para treinar um pouco mais sobre as animações com JavaScript.

- **window.requestAnimationFrame()**, disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/API/Window/requestAnimationFrame>.

O site Manual do Código tem um tutorial em texto e vídeo com um exercício bem desafiador usando animações com o Canvas. Acesse, veja a animação que eles criaram e tente fazer igual.

- **Tutorial Html5 CANVAS. Parte 2. Animações**, disponível em: <https://www.manualdocodigo.com.br/tutorial-canvas-parte2/>.

Referência Bibliográfica

MDN Web Docs. **Canvas tutorial.** Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/API/Canvas_API/Tutorial>.

SOUZA, R. F. M. **Canvas HTML5: composição gráfica e interatividade na web.** Brasport: Rio de Janeiro. 2013.

Ir para exercício